Attorney Docket No. CISCP110C1

PATENT APPLICATION

LOAD SHARING AND REDUNDANCY SCHEME

Inventors: Bhushan Mangesh Kanekar
4022 Chamberer Drive
San Jose, CA 95135
Citizenship: India

Saravanakumar Rajendran
3308 Provence Court
San Jose, CA 95135
Citizenship: India

Jonathan Davar
26541 Purissima Road
Los Altos Hills, CA 94022
Citizenship: USA

Assignee: Cisco Technology, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

A corporation of California

Prepared by:
BEYER & WEAVER, LLP
P.O. Box 778
Berkeley, CA 94704
Telephone (510) 843-6200

LOAD SHARING AND REDUNDANCY SCHEME

RELATED APPLICATIONS

This application is a continuation of Patent Application No. 09/342,859, Attorney Docket No. CISCP110, entitled "Load Sharing and Redundancy Scheme," naming Kanekar et al. as inventors, filed on June 29, 1999, which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to networking technology. More particularly, the present invention relates to providing load sharing and redundancy in a network through a master router and a slave router having a shared set of interfaces in a single device.

2. Description of the Related Art

Networks are commonly used by organizations for a variety of purposes. For instance, through the use of networks, resources such as programs and data may be shared by users of the network. In addition, a computer network can serve as a powerful

communication medium among widely separated users.

Communication among hosts and users of a network is often facilitated through connection to one or more routers. As shown in FIG. 1, a host 112 normally resides on a network segment 114 that enables its network entities to communicate with other entities or networks. Note that the host 112 need not directly connect to the entities or networks with which it communicates. For example, as shown in FIG. 1, the host 112 may be connected through a router R1 116. The router R1 116 may, in turn, connect one or more other routers such as router R2 118 with selected entities or networks.

Now, suppose that the host 112 wishes to send a message to a corresponding node 120. A message from the host 112 is then packetized and forwarded through the appropriate routers and to the corresponding node 120, as indicated by dotted line "packet from host"122, according to a standard protocol. If the corresponding node 120 wishes to send a message to the host 112 – whether in reply to a message from the host 112 or for any other reason – it addresses that message to the IP address of the host 112 on the network segment 114. The packets of that message are then forwarded to router R1 116 and ultimately to the host 112 as indicated by dotted line "packet to host"124.

As described above, packets sent to and from the host 112 are forwarded via the router R1 116. As shown, the router R1 116 is the only route to and from the host 112. Thus, if the router R1 116 fails, communication with the host 112 becomes impossible. Accordingly, the reliability of the network as well as the routers in the network is of utmost importance.

As networks become a critical resource in many organizations, it is important that the networks are reliable. One way of achieving reliability is through redundancy. As

described above, a single router failure may prevent communication to and from each host and user connected to the router. In many networks, it is common to provide redundancy through the use of multiple routers such that a backup router functions in the event of failure of a primary router. However, when the primary router fails, there is typically a "switchover time" that is required for the backup router to take over the functions of the primary router. As a result, such attempts to provide redundancy in switches suffer from a large switchover time. Accordingly, it would be beneficial if such redundancy could be provided with a reduction in the switchover time from a non-functional to a functional router.

In addition to reliability, it is often desirable to improve performance within a given network. In order to achieve this improvement, load sharing is often preferable. For instance, various users of a network may have a higher traffic level than other users of the network. It would therefore be desirable if performance could be achieved through the distribution of traffic among multiple routers.

In view of the above, it would be desirable if a redundancy and load sharing scheme could be implemented to reduce the switchover time upon failure of a router while implementing a load sharing scheme among multiple routers operating in a single device.

5

SUMMARY OF THE INVENTION

An invention is described herein that provides load sharing and redundancy in a network. This is accomplished, according to one embodiment, through the use of a master router and a slave router operating in the same chassis and having a shared set of interfaces. Prior to failure of the master router, the master router communicates shared state information to the slave router. In addition, the slave router operates in "standby mode" to obtain information from the shared set of interfaces. In this manner, the switchover time required to switch from the master router to the slave router upon failure of the master router is significantly reduced.

According to one aspect of the invention, a default gateway is associated with both the master router and the slave router. This is accomplished by assigning a shared IP address and a shared MAC address to both a first router and a second router so that the shared IP and MAC addresses are shared between the first router and the second router. Additionally, a first MAC address is assigned to the first router and a second MAC address is assigned to the second router. The default gateway is configured on the hosts such that a default gateway IP address is associated with the shared IP address. The shared IP and MAC addresses are associated with one of the routers (e.g., the first router or master router). When the master fails, the slave takes over both the shared IP address and the shared MAC address.

In order to route traffic, there are three layers of protocol: a physical layer, a data link layer, and a network layer. The data link layer is often referred to as "layer 2" while the network layer is often referred to as "layer 3." The responsibility of the data link layer is to transmit chunks of information across a link. The responsibility of the network layer

is to enable systems in the network to communicate with each other. Thus, the network layer finds a path or "shortcut" through a series of connected nodes that must forward packets in the specified direction.

According to another aspect, the master and the slave each includes a switching processor to switch packets in hardware and a routing processor to enable packets to be routed in software. The switching processor is adapted for running a layer 2 protocol (e.g., spanning tree) and the routing processor is adapted for running a layer 3 routing protocol. In addition, the master and the slave each maintains its own forwarding data. More particularly, the master and the slave each maintain a layer 2 database associated with the layer 2 protocol and a routing table associated with the layer 3 routing protocol. Both the master and the slave independently run its own layer 3 routing protocol and maintain its own routing table. However, only the master runs the layer 2 protocol. More particularly, the master saves the layer 2 protocol information in a layer 2 protocol database (e.g., spanning tree database) and sends layer 2 protocol updates to the slave so that it may similarly store the layer 2 protocol updates in its own layer 2 protocol database. When the master fails, the slave then runs the layer 2 protocol and accesses its own layer 2 protocol database. Since the slave maintains its own layer 2 protocol database and layer 3 routing table, switchover time upon failure of the master is minimized.

According to another aspect, prior to failure of the master, the slave receives updates from the master in order to synchronize operation of the two routers. For instance, the master maintains the hardware information for both the master and the slave. Therefore, in addition to sending layer 2 protocol updates, the master also sends other information related to the hardware shared by the two routers. As one example, multicast

group membership for the shared ports is sent by the master to the slave. As another example, hardware information such as temperature and information related to the power supply is sent by the master to the slave.

According to yet another aspect, the master and the slave each include a forwarding engine in addition to the routing processor and the switching processor. The forwarding engines are adapted for forwarding packets in hardware and therefore increase the speed with which packets are forwarded. Each forwarding engine has an associated set of forwarding engine tables. More particularly, each forwarding engine includes a layer 2 table associating each destination MAC address with a port and router. Thus, if a packet cannot be forwarded in hardware or it is undesirable to forward the packet in hardware, the packet is forwarded by the router specified in the layer 2 table. In addition, a layer 3 shortcut table stores shortcuts (i.e., layer 3 forwarding information) for a path from a particular source IP address to a particular destination IP address. When a router forwards a packet, a shortcut is created and entered in the layer 3 shortcut table. Packets may then be forwarded by the forwarding engine for this particular path.

According to another aspect, the slave operates to update its forwarding tables during standby mode as well as upon failure of the master. In order for the slave to forward a packet, the layer 2 table of the slave's forwarding engine must contain an entry associating the desired destination MAC address with the slave router. Moreover, for the forwarding engine (i.e., hardware) of the slave to forward a packet, there must be an entry for the particular path from the source IP address to the destination IP address. Thus, prior to failure of the master, the slave's forwarding engine observes packets at the shared interfaces to obtain information from the packet header to establish shortcuts. For

instance, the slave may obtain a shortcut established by the master from the packet header. The slave then updates its layer 2 and layer 3 tables with an appropriate entry as necessary.

Upon failure of the master router, the slave modifies its forwarding engine tables to enable packets to be forwarded by the slave. At a minimum, in order to forward packets in software, the slave's layer 2 table is modified to associate destination MAC addresses with the slave rather than the master. In addition, in order for a packet to be forwarded via the forwarding engine (i.e., hardware) of the slave, an entry for the specific path is identified in the slave's layer 3 table. Thus, if an entry exists in the slave's layer 3 table for the flow (e.g., path from source to destination) as provided in the packet header, the packet may be forwarded by the forwarding engine. Even if the entry in the slave's layer 3 table for that particular flow is not modified by the slave, packets may be forwarded using information in the current entry using the shortcut established by the master (e.g., with the source MAC address identifying the master). However, it is desirable to forward packets with the correct source MAC address (e.g., the MAC address of the slave). According to one embodiment, since the master and the slave routers may potentially arrive at different routing decisions and therefore different shortcuts, the shortcuts established by the master are invalidated. In order to invalidate these shortcuts, they are removed from the slave's layer 3 shortcut table. However, if all shortcuts are removed simultaneously, a large number of packets will need to be forwarded in software. Therefore, entries in the slave's layer 3 shortcut table are selected and removed gradually. For example, the entries may be removed according to port number or other criteria. Once a packet is forwarded by the slave router in software, a correct entry is created and entered in the slave's shortcut table. Packets may then be forwarded by the slave with a current shortcut as well as correct source MAC address. Thus, since the slave maintains its own forwarding engine tables,

packets may be forwarded with a minimum delay time.

According to another aspect, the configuration of the master and the slave is synchronized. There are three categories of information that may be configured for each router. First, there is information that must be the same for both routers. Second, there is information that must be different for both routers. Third, there is information that can be different but is recommended to be the same for both routers. Thus, the same configuration file may be maintained on both the master and the slave to enable the routers to be synchronized with these three categories of information.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram illustrating communication between a host and a corresponding node via a router.

FIG. 2 is a diagram illustrating a system in which multiple routers are used to provide redundancy.

FIG. 3 is a general block diagram illustrating routers that share a single set of interfaces according to an embodiment of the invention.

FIG. 4 is a diagram illustrating an exemplary configuration file according to an embodiment of the invention.

FIG. 5 is a diagram illustrating a routing and switching system according to one embodiment of the invention.

FIG. 6 is a process flow diagram illustrating one method of determining which router is the master.

FIG. 7 is a block diagram illustrating a VLAN in which multiple LANs are grouped together.

FIG. 8 illustrates an exemplary system for load sharing using VLANs according to an embodiment of the invention.

FIG. 9 is a block diagram illustrating a database configuration for the routing and switching system according to an embodiment of the invention.

FIG. 10 is a process flow diagram illustrating one method of configuring the master and slave routers at start up.

FIG. 11A is a process flow diagram illustrating one method of operating the master and slave prior to failure of one of the routers according to one embodiment of the invention.

FIG. 11B is a process flow diagram illustrating one method of forwarding packets prior to failover.

FIG. 12A is a process flow diagram illustrating one method of transitioning to the slave upon failure of the master according to an embodiment of the invention.

FIG. 12B is a process flow diagram illustrating one method of operating upon

failure of the slave according to an embodiment of the invention.

FIG. 12C is a process flow diagram illustrating one method of modifying the forwarding engine tables of the slave after failure of the master according to an embodiment of the invention.

FIG. 12D is a process flow diagram illustrating one method of forwarding packets by the slave as shown at block 1212 of FIG. 12B after failure of the master according to an embodiment of the invention.

FIG. 13A is a diagram illustrating an exemplary layer 2 table that may be independently maintained by the master and the slave.

FIG. 13B is a diagram illustrating an exemplary layer 3 table that may be independently maintained by the master and the slave router.

FIG. 14A is a diagram illustrating the need for second hop redundancy within a network.

FIG. 14B is a diagram illustrating the problem created when second hop redundancy is not provided.

FIG. 15 is a block diagram of a network device that may be configured to implement aspects of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following description, numerous specific details are set forth in order to

provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without some or all of these specific details. In other instances, well known process steps have not been described in detail in order not to unnecessarily obscure the present invention.

There are several ways to provide redundancy using multiple routers. For instance, two separate fully operational routers are often used to provide redundancy. FIG. 2 is a diagram illustrating such a system. As shown, a first router R1 102 and a second router R2 104 are supplied to provide redundancy in a system supporting client 106. The first router R1 102 and the second router R2 104 share a common interface 108. In addition, the first router R1 102 has an associated set of interfaces 110 and the second router R2 104 has a separate set of interfaces 112. Thus, redundancy is commonly implemented to ensure that the client 106 is supported in the event that one of the routers 102 and 104 fails. It is important to note that where multiple routers are used, they typically do not share the same interfaces. As a result, the configurations of the routers cannot be identical. Moreover, since the two routers are separate, the time to detect a failure of one of the routers is substantial.

In order to reduce the time required to detect a failure of one of the routers, the two routers may be provided in the same chassis. However, the interfaces are typically not easily shared between two routers. As a result, the configuration information cannot be shared between the routers. Moreover, since the interfaces are not shared, both routers cannot be fully operational. Since both routers are not fully operational, when one of the routers fails, there is often a substantial "switchover time" during which the alternate router is brought up. More particularly, during this time, the appropriate software is

downloaded to the secondary router to enable the secondary router to take over the interfaces associated with the primary router.

As described above, although multiple routers are commonly used to provide redundancy in a network, the routers do not typically share a set of interfaces. As a result, the configurations of the routers are not identical and therefore the switchover time in the event of failure of the primary router (i.e., master) may be substantial. To solve this problem, the present invention provides at least two routers that share the same set of interfaces. More particularly, since both routers share the same set of interfaces, both the routers may be fully operational. FIG. 3 is a general block diagram illustrating two routers provided in the same chassis that share a single set of interfaces according to an embodiment of the invention. As shown, a first router 202 and a second router 204 share a set of interfaces 206-1, 206-2, and 206-3. Since the routers 202 and 204 share the same set of interfaces, the routers may share at least some configuration information 208 and therefore the switchover time as experienced by client 210 may be reduced. Since the routers 202 and 204 are in the same chassis 212, it is easier to ensure a similar configuration on both routers. For instance, both routers may be configured via a single console. In addition to sharing configuration information, the routers may each provide updates to the other router. For instance, where R1 202 is the master and R2 204 is the slave, information such as spanning tree protocol updates may be sent from the master to the slave as shown at 214.

According to one embodiment, two independent routers function in the same chassis to seamlessly forward packets through the use of the Hot Standby Redundancy Protocol (HSRP). According to the Hot Standby Redundancy Protocol (HSRP), a protocol

available from Cisco Systems, Inc. located in San Jose, California, the master router and the slave router share a common MAC address and IP address. In addition, each of the routers has its own unique MAC address that will be used by the router for advertising routes to other routers. One of the routers acts as the master and it responds to Address Resolution Protocol (ARP) queries for the shared IP address with the shared MAC address. The default gateway may be configured by associating a default gateway IP address to the shared IP address. The IP to MAC binding may be either statically configured or obtained through the ARP protocol. When the master fails, the slave takes over both the shared IP address and the shared MAC address that was owned by the master. Thus, a client need only know the default gateway IP to route packets.

In order to configure the routers, there are three categories of information that may be configured for each router. First, there is information that must be the same for both routers. Second, there is information that must be different for both routers. Third, there is information that can be different but is recommended to be the same for both routers. One desirable configuration for a set of routers having the same interfaces is described as follows. More particularly, the configurations that need to be the same include the number of ports in each line card (i.e., router), the type of ports (e.g., type of VLAN to which each port belongs), and security information (e.g., access lists). Configurations that need to be different include the IP addresses associated with each interface of the routers. In other words, multiple routers cannot have the same IP address for a particular interface. In addition, the priorities associated with each router are different in order to enable load sharing among the different routers. Configurations that can be different but are recommended to be the same include routing protocols and routing tables associated with the routers. One method of implementing load sharing is described below with reference

to FIGS. 8A and 8B.

According to a specific embodiment, in order to provide the configuration information for the routers sharing the same set of interfaces, a shared configuration file is created. FIG. 4 illustrates an exemplary shared configuration file. As shown, the configuration file 402 includes configuration information in multiple command lines. The command lines may be stored as text strings, as shown. Alternatively, the command lines may be stored in a binary format. The configuration may be stored in non-volatile RAM such that when read, the routers may have all necessary information to operate. More particularly, as shown, each command line 404 identifies a particular configuration with a specified router (e.g., router R1 or R2). In addition, where the configurations for the routers are different, when one router is configured, the configuration for the second router is forced. As shown, this may be accomplished by configuring both routers on the same command line. By way of example, where the IP address 406 (e.g., 1.1.1.2) and associated mask 410 are configured for a specified "master" router 412, an alternate "slave" router 414 may simultaneously be configured with an IP address 416 (e.g., 1.1.1.3) and a mask 418. More particularly, the two IP addresses 406 and 416 must be in the same subnet.

A more detailed diagram illustrating a routing and switching system according to one embodiment of the invention is presented in FIG. 5. As shown, in this embodiment, two independent operational routers R1 502 and R2 504 are provided in a single chassis, permitting the routers to communicate in the backplane of the device. When redundancy is implemented using HSRP, routers communicate their existence through "hello" packets. Thus, a failure of one of the routers is detected by another router when a specified number of consecutive "hello" packets are not received during a period of time. Since the routers

communicate in the backplane of the device, a failure of one of the routers may be detected in hardware. As a result, the time in which a failure of one of the routers may be detected is minimized. Each of the routers 502 and 504 is shown in detail as including a corresponding routing processor 506 and 508, switch processor 510 and 512, and forwarding engine 514 and 516, respectively. More particularly, the routing processors 506 and 508 run the layer 3 routing protocols. In addition, since the device also functions as a bridge, the switch processors 510 and 512 are adapted for handling the layer 2 protocols (e.g., spanning tree protocol) and may therefore control the hardware by initializing the associated forwarding engines 514 and 516. However, since only one spanning tree will be used, only one of the switch processors runs the layer 2 spanning tree protocol. Therefore, the master runs the layer 2 spanning tree protocol until the master fails, at which time the slave starts running the layer 2 spanning tree protocol. The forwarding engines 514 and 516 may perform forwarding in hardware and therefore each functions as a switch.

The slave maintains its own backup information, including layer 2 and layer 3 tables. More particularly, the slave operates in standby mode and therefore obtains information by observing packets as they are received at the interfaces shared with the master. In addition, the master sends selected information to the slave during normal operation of the master, as shown at 518. For instance, when the layer 2 spanning tree protocol is updated by the master, these spanning tree protocol updates are communicated to the slave. Both the master and slave each maintain its own spanning tree database. Although only the master runs the spanning tree protocol, the slave receives the spanning tree updates from the master and stores the spanning tree updates in its own spanning tree database. As a result, the master and the slave have identical spanning tree databases,

thereby providing layer 2 redundancy. Although layer 2 information is shared, information in layer 3 (i.e., routing tables) is not dynamically shared between the routers (e.g., by the routing processors), and therefore each router maintains a separate routing table. In addition, each of the forwarding engines 514 and 516 maintains its own tables, which will be described below with reference to FIGS. 13A through 13C. Thus, the slave maintains its own forwarding engine tables, spanning tree database, and routing table. Since the slave maintains its own tables and receives information such as spanning tree updates from the master, switchover time is minimized upon failure of the master.

One of the routers may fail under a variety of circumstances. First, the routing processor of one of the routers may fail. Second, one of the switch processors may fail. Third, one of the forwarding engines may fail. According to one embodiment, any of these and other failures are treated as a failure of the entire router.

While both routers are fully operational, only one functions as the "master" while the other functions as the "slave." The master therefore actively forwards packets while the slave functions in standby mode. When the master fails, the slave takes over to forward any remaining packets. During initialization of the routing system, one of the routers is determined to be the "master". A process flow diagram illustrating one method of determining which router is the master is presented in FIG. 6. Initialization of the routers as either master or slave begins at block 600. At block 602, one of the routers receives a signal from the other router indicating which router is the master router. According to one embodiment, the master sends a signal to the slave to assert that it is the master. For instance, the first router to come up may assert such a signal. This is desirable since the first router to come up will have a greater capacity for handling incoming and

outgoing packets. If both routers come up simultaneously, a priority previously assigned to the routers may be used to determine which router will function as the master. For instance, the priority associated with each router may be set up by a Network Administrator. If at block 604 it is determined that the signal asserts that the sending router is the master router, the receiving router (i.e., the router receiving the signal) is determined to be the slave at block 606. Otherwise, the receiving router is determined to be the master at block 608.

One or more default gateways may be configured using Hot Standby Redundancy Protocol (HSRP) / Multigroup HSRP (M-HSRP) such that the master will be responsible for routing packets from a subset of interfaces and the slave will be responsible for routing packets from the remaining interfaces. HSRP/M-HSRP is a protocol available from Cisco Systems, Inc. located in San Jose, California that provides a redundancy mechanism when more than one router is connected to the same segment of a network (e.g., Ethernet, FDDI, Token Ring). The participating routers share a common predefined MAC address and IP address. In addition, each of the routers has its own unique MAC address which will be used by the router for advertising routes to other routers. In addition, this unique MAC address will be used as the source MAC address portion of the routed packets. One of the participating routers acts as the Master and it responds to Address Resolution Protocol (ARP) queries for the shared IP address with the shared MAC address. The default gateway may be configured by associating a default gateway IP address to the shared IP address and the IP to MAC binding may be either statically configured or obtained through the ARP protocol. When the master fails, the slave takes over both the shared IP address and the shared MAC address that was owned by the master. In this manner, the slave takes over the master's interfaces upon failure of the master. Thus, a host (i.e., client)

need only know the default gateway IP to route packets. As a result, the client need not be aware of which router is the master router. Nor must the client be notified when one of the routers fails.

While one default gateway may be used, it may also be desirable to use a different default gateway for different groups of users. For instance, it may be desirable to configure a first set of users to use a first default gateway and a second set of users to use a second default gateway. One way to logically group users together is through the use of virtual LANs (VLANs). FIG. 7 is a block diagram illustrating a VLAN in which multiple LANs are grouped together. As shown, router R1 702 has a plurality of interfaces that may connect to multiple LANs. As shown in FIG. 7, a first LAN 704 is coupled to a first interface 706, a second LAN 708 is coupled to a second interface 710, a third LAN 712 is coupled to a third interface 714, and a fourth LAN 716 is coupled to a fourth interface 718. As shown in FIG. 7, the first LAN 704 and the second LAN 708 are coupled into first VLAN 724 while the third LAN 712 and the fourth LAN 716 are coupled into second VLAN 728. Bridging is used to communicate among LANs of the same VLAN while routing is used to communicate across VLANs. In this manner, LANs may be grouped together according to various criteria such as functionality or project.

As described above, packets are routed across VLANs. FIG. 8 illustrates an exemplary system for load sharing using VLANs according to an embodiment of the invention. Routers R1 800 and R2 802 are both shown to have two interfaces, a first interface 804 and a second interface 810. The first interface 804 is connected to network segments 1.1.1.2 and 1.1.1.3 to a first VLAN 808. Similarly, the second interface 810 is connected to network segments 2.2.2.2 and 2.2.2.3 to a second VLAN 814. Multiple

VLANs and clients 812 may be active on the same interface. As a result, it may be desirable to distribute the load among the multiple routers (i.e., R1 800 and R2 802) as well as accommodate the different traffic levels of each user. Thus, the requirements of each VLAN may be met through load distribution among the routers R1 800 and R2 802. Moreover, the load may be distributed among the routers R1 800 and R2 802 based upon the source of incoming packets to the routing system. More particularly, this may be accomplished through associating the users and/or VLANs with different default gateways. In this manner, the unique requirements of the different VLAN groups as well as the users within each group may be separately accommodated. Accordingly, load sharing can be achieved by configuring multiple groups between the master and slave routers and thus directing traffic to both the routers.

As described above, the load (e.g., incoming load) may be distributed among the master and the slave. For instance, suppose clients on VLAN 1 and VLAN 2 have their default gateway configured to be the master and clients on VLAN 3 and VLAN 4 have their default gateway configured to be the slave. More particularly, the hosts in VLAN 1 and VLAN 2 are configured with a default gateway of the default gateway IP address for that group and the corresponding MAC address. Similarly, the hosts on VLAN 3 and VLAN 4 are configured with the slave's information. When one of the two routers fails, the other router takes over the hosts serviced by the other router. For instance, when the master fails, the slave services the hosts on VLANs 1 and 2 in addition to the hosts on VLANs 3 and 4. Moreover, since the slave is already a member of VLANs 1 and 2 as a separate router, it already has the appropriate routing information and therefore does not have to recalculate any routing tables.

As described above with reference to FIG. 6, it is initially determined which one of the routers is master. The routers may be prioritized to provide redundancy as described above with reference to FIG. 6. In addition, load sharing may be implemented using default gateways as described above with reference to FIGS. 7 and 8. In order for the routing and switching system to function in the event of failure of one of the routers, the system is configured such that switchover time is minimized. This is accomplished, in part, through the maintaining and updating of information for both the master and slave during normal operation of the master. As a result, when the master fails, the time required to bring up the slave is minimized.

Typically, in a routing and switching system, the hardware and software maintains layer 2 and layer 3 information in order to forward packets. According to one embodiment of the invention, each of the routers and forwarding engines maintains its own layer 2 and layer 3 data. As shown in FIG. 9, although the routers are provided in a single chassis, each of the routers 902 and 904 has its own layer 3 routing table 906 and 908 as well as its own layer 2 spanning tree database 910 and 912, respectively. In addition, associated forwarding engines 914 and 916 each maintain a set of forwarding engine tables. As shown, the first forwarding engine 914 has a set of forwarding engine tables 918 including a layer 2 table 920 and layer 3 shortcut table 922. In addition, the second forwarding engine 916 has a set of forwarding engine tables 924 including a layer 2 table 926 and layer 3 shortcut table 928. Where the first router 902 is the master and the second router 904 is the slave, the master sends information such as spanning tree updates to the slave, as shown at line 930. In addition, the slave and master routers 902 and 904 each maintains its own routing table 906 and 908, respectively, through routing updates received from other routers in the network. Similarly, each forwarding engine 914 and 916 updates its

associated forwarding engine tables 918 and 924, respectively, through information obtained from packet headers of packets observed at the shared interfaces (not shown to simplify illustration).

In addition to determining which router is the master, both routers must be brought up such that they are fully functional. One method of configuring the master and slave routers at start up is presented in FIG. 10. As shown, the process at start up begins at block 1000 and the routers are brought up at block 1002. The configuration information is read from the configuration file for both routers at block 1004. As described earlier, the configuration information may fall in one of three categories. The routers start running the routing protocols at block 1006. The routing protocols of the two routers may be different, but are recommended to be the same. Each router then builds its own routing table as shown at block 1008. The routing tables are not synchronized. As known to those of ordinary skill in the art, the routers dynamically exchange routing updates. Each router then updates its own routing table based upon the information gathered in each of the routing updates.

Once both routers are fully functional, the master and slave continue to communicate information prior to failure of one of the routers. As shown in FIG. 11A, a method of operating the master and slave prior to failure of one of the routers according to one embodiment of the invention is presented. The process begins at block 1100 and at block 1102, a synchronization task runs in master/slave mode and the master sends to the slave synchronized state information to synchronize the port states and forward delay time. By way of example, the state of each port may indicate that the link is up or down, that the port is blocked, listening, or forwarding.

Both the master and the slave run layer 3 routing protocols and therefore each maintains its own routing table. However, only one of the routers runs the layer 2 spanning tree protocol at any given point in time. More particularly, prior to failure of the master router, the master runs the layer 2 spanning tree protocol. Only upon failover of the master router does the slave router run the layer 2 spanning tree protocol. Thus, at block 1104, the master sends a spanning tree update to the slave (e.g., specifying spanning tree states). For instance, the spanning tree update may indicate the states of the ports. Next, at block 1106, the slave acknowledges the spanning tree updates. The slave then updates its own spanning tree database such that the slave's spanning tree database is substantially identical to that maintained by the master. In addition, the VLAN membership of the master is sent to the slave at block 1108. In this manner, the slave may quickly determine which VLANs it will be supporting when the master fails. Forwarding engine information is then sent by the master to the slave to initialize the hardware of the slave at block 1110. Forwarding engine information may include, but is not limited to, port membership (i.e., association between ports and receivers), multicast group membership (e.g., which ports are members of which multicast groups). In addition, hardware information may be sent as necessary by the master to the slave at block 1112. Hardware information may include, but is not limited to, temperature and indication of power supply failure.

FIG. 11A describes a method of operating the master router prior to failure of the master router. In addition, when a packet is received at the shared set of interfaces and forwarded by the master, the forwarding engine tables are updated by both the master and the slave. One method of forwarding packets prior to failover is presented in FIG. 11B. The process begins at block 1116 and at block 1118, the master receives a packet at the shared set of interfaces. Thus, the master obtains information from the packet header

while actively forwarding the packet. Although the master may send this information to the slave via software, this is a time-consuming process. Since it is necessary for the slave to obtain the information required for its forwarding engine tables in a less time-consuming manner, the slave operates during "standby mode" to observe incoming and outgoing packets at the set of shared interfaces. Thus, the slave independently obtains information from the packet observed at the shared set of interfaces at block 1120. The master then updates the master's forwarding engine tables at block 1122, as necessary, with an entry associated with the packet. Exemplary forwarding engine tables will be shown and described with reference to FIGS. 13A and 13B. Similarly, at block 1124 the slave updates the slave's forwarding engine tables as necessary with an entry associated with the packet. The master then forwards the packet at block 1126. Therefore, at any given point in time, both the slave and the master will have essentially identical forwarding engine tables.

As described above, according to one embodiment, a failure of the hardware (i.e., switching engine) or software (i.e., routing processor or switch processor) in a router is treated as a failure of the entire router. FIG. 12A is a process flow diagram illustrating a method of transitioning to the slave upon failure of the master according to one embodiment of the invention. Upon failure of the master at 1200, a backplane signal is sent to the slave at block 1202. The slave then starts the layer 2 spanning tree protocols at block 1204. At block 1206, the slave then uses the synchronized state information previously sent by the master to the slave as shown at block 1102 of FIG. 11A. It is important to note that the slave typically starts at ground zero to obtain such synchronized state information. Since the slave need not start from ground zero, the failover time is substantially reduced.

As described above, in order to provide load sharing in the routing system, certain interfaces may have a specified default gateway (e.g., R1). Thus, when R1 fails, R2 must be specified as the new default gateway so that the forwarding engine tables may be modified accordingly. Exemplary forwarding engine tables and mechanisms for modifying these tables will be described in further detail below with reference to FIGS. 12C, 13A and 13B. Thus, at block 1208, the routing processor of the slave sends a signal to the forwarding engine to replace the references to the MAC address and IP address of the master with the MAC address and IP address of the slave, where appropriate. The forwarding engine tables of the slave are then modified at block 1210 so that packets may then be forwarded by the slave router at block 1212. An exemplary method of modifying the forwarding engine tables will be described with reference to FIG. 12C and exemplary forwarding engine tables will be described in further detail with reference to FIGS. 13A and 13B.

When the slave fails, the slave merely notifies the master of its failure. As shown in FIG. 12B, when the slave fails 1220, a signal is sent to the master at block 1222.

Packets received at the shared interfaces may be forwarded in hardware via the forwarding engine or in software. However, packets must be encapsulated in the same manner regardless of whether the packets are forwarded in hardware or software. Thus, similarly to the information maintained by the routing processor and switching processor, the forwarding engines maintain layer 2 and layer 3 tables, as will be shown and described with reference to FIGS. 13A and 13B. As shown at block 1210 of FIG. 12A, the forwarding engine tables of the slave are modified after failure of the master to enable packets to be accurately forwarded by the slave. One method of modifying the forwarding

engine tables of the slave after failure of the master is presented in FIG. 12C. The process begins at block 1230. In the absence of a layer 3 entry for a particular flow, a packet following this flow is sent via software. In order to determine the router used to send the packet, the layer 2 table is used to match the destination MAC address of the packet and therefore must contain updated information. Thus, at block 1232, entries in the slave's layer 2 table that are associated with the master are modified or replaced such that the resulting entries are mapped to the slave rather than the master. Once modified, the slave's layer 2 table may be used to determine the appropriate router to forward packets in the absence of an entry in the slave's layer 3 table. During forwarding of a packet, if there is no layer 3 entry, an entry in the layer 2 table associated with the destination MAC address of the packet is identified. The router identified in this layer 2 table entry is then used to forward the packet in software until a layer 3 entry for this flow is created.

In addition, the slave's layer 3 shortcut table is modified. Since the slave and the master share the same interfaces and are independently running routing protocols, they both should arrive at the same routing decision for a particular IP destination. However, there is no guarantee that all the routing updates will reach and get processed by both the slave and the master all the time. In theory, both the master and the slave will come to the same routing decisions. In addition, shortcuts in the router's layer 3 table are established upon forwarding of a packet by the router based upon information in its routing table. However, since the slave and the master operate independently, the shortcuts cannot be guaranteed to be identical for both the master and the slave. Moreover, these potentially invalid shortcuts take up space in a limited amount of space in the layer 3 table in hardware. Therefore, the shortcuts created by the master are invalidated on failover. As a result, at block 1234, selected entries associated with the master are removed from the

slave's layer 3 table. Prior to removal of the entries from the slave's layer 3 table, packets may be routed via the slave's forwarding engine using the master's MAC address as the source MAC address. Once an entry for a particular flow is removed, packets may be forwarded in software until a new entry for the flow is created in the slave's layer 3 table. Later, when a packet belonging to the same flow (e.g., from the source IP address to the destination IP address) is routed by the slave, this removed entry is effectively "replaced" with an entry associated with the slave for this same "flow." Once the entry is replaced, packets may be routed via the slave's forwarding engine using the slave's MAC address as the source MAC address. In addition, on switchover, the floating default gateway IP address and the associated floating MAC address is now associated with the slave (e.g., with the MAC address of the slave). Accordingly, in order to enable forwarding by the slave's forwarding engine upon failure of the master without a period of delay, the shortcuts created by the master are used in the interim period after failure of the master and prior to updating the slave's layer 3 shortcuts.

As described above with reference to FIG. 12C, packets may be forwarded by the slave router in hardware or software. Moreover, when the packet is forwarded in hardware (by the slave's forwarding engine) the source MAC address may be that of the master or the slave depending upon the status of the slave's forwarding engine tables. A flow diagram illustrating one method of forwarding packets by the slave as shown at block 1212 of FIG. 12B after failover of the master is presented in FIG. 12D. The process begins at block 1240 and at block 1242, the slave determines whether the packet is to be forwarded in software. For instance, even after the slave's layer 2 table has been modified, if there is no entry in the slave's layer 3 table for the path specified by the packet's header, the packet is forwarded in software. Thus, it is determined whether an entry associated

with the packet is present in the layer 3 table. More particularly, using information in the packet header, it is determined whether the layer 3 table includes an entry associated with the source IP address and the destination IP address of the packet. If the layer 3 table does not include such an entry, the packet is routed in software until an entry has been created for the specified flow (i.e., path from source to destination). Moreover, if forwarding of the packet requires extra processing that cannot be performed or is difficult to perform in hardware, the packet is forwarded in software. The packet is then forwarded in software at block 1244 using the slave's routing tables and spanning tree protocol database. Once the packet is routed, an entry is created in the layer 3 table such that the source MAC address is that of the slave at block 1246.

As described above, if the slave's layer 3 table does not include an entry associated with the packet or it would otherwise be difficult or impossible to forward the packet in hardware, the packet is forwarded in software. Otherwise, the packet is routed via the forwarding engine and the process continues at block 1248 where it is determined whether the slave's layer 3 table includes a new or modified entry associated with the path of the packet to be forwarded. The packet is then forwarded with the appropriate source MAC address and destination MAC address as specified by the entry in the layer 3 table. More particularly, if the layer 3 table contains an entry that has not been removed or modified by the slave, the source MAC address identifies the master. However, if the layer 3 table includes an entry that has been created or modified by the slave, the source MAC address identifies the slave. Thus, if it is determined at block 1248 that the slave's layer 3 table does not include a new entry created by the slave, the packet is forwarded via the forwarding engine using the slave's forwarding engine tables and the source MAC address of the master at block 1250. If the slave's layer 3 table does include a new entry created

by the slave, the packet is forwarded via the forwarding engine using the slave's forwarding engine tables and the source MAC address of the slave at block 1252.

Since both the slave and the master are independent operational routers, they may each come to different routing decisions. As a result, the slave and the master each maintains its own set of forwarding engine tables. Since the slave and the master share the same set of interfaces, the slave may observe incoming and outgoing packets and therefore obtains information to update its layer 2 and layer 3 tables. More particularly, prior to failure of the master, the master monitors all traffic entering the switch during active forwarding of packets while the slave monitors all traffic entering the switch while the slave is in standby mode. Thus, while the master's forwarding engine is actively forwarding packets, the slave is learning information from the bus (e.g., incoming packets). Once the master fails, the slave actively forwards packets and monitors all traffic coming into the switch, as the master did prior to its failure.

Exemplary forwarding engine tables are described with reference to FIGS. 13A and 13B. More particularly, FIG. 13A is a diagram illustrating an exemplary layer 2 table that may be independently maintained by the master and the slave. The layer 2 table serves as a bridge forwarding database and therefore is used to determine the LAN and port used to send packets out. During normal operation, prior to failure of the master, both the slave and the master each monitor all traffic coming into the switch. Based upon the header of the incoming packet, an entry in the corresponding layer 2 table is created. As shown, the layer 2 table 1302 specifies a MAC address 1304 of a host as specified by the source MAC address of the incoming packet, an associated VLAN 1306 to which the host belongs, and a port 1308 that the packet has come in on. In addition, each entry is associated with a

router 1309 (e.g., identified with the destination MAC address of the incoming packet), which may be accomplished in the layer 2 table or in a separate mapping table. For instance, the router 1309 may specify the slave or the master router. However, upon failure of the master, the slave modifies its layer 2 table entries to specify the slave as the router. When an entry for a particular flow is not in the layer 3 table, the packet is routed via a router associated with that flow. More particularly, an entry in the layer 2 table (or a separate mapping table) is matched with the destination MAC address as specified in the packet header. In this manner, the outgoing VLAN and outgoing port for a specified destination MAC address may be obtained from information learned from previously received incoming packets.

In addition, the master and the slave router each maintains its own layer 3 shortcut table. FIG. 13B is a diagram illustrating an exemplary layer 3 table 1310 that may be maintained by the master and the slave router. As shown, each entry in the L3 routing table specifies a destination IP address 1312, a source IP address 1314, a destination MAC address 1316, and a source MAC address 1318. As described above, since the slave and master share a single set of interfaces and therefore the same packet information, when a packet is forwarded by the master (e.g., by the routing processor), a shortcut is established and a corresponding entry is entered into the layer 3 table of the slave as well as that of the master. More particularly, the slave obtains the shortcut previously established by the master from the packet header. Prior to failure of the master, the slave's forwarding engine is in standby mode. During the standby mode, the slave's forwarding engine learns information from the bus (e.g., from the headers of the incoming packets) and updates its layer 3 shortcut table. As a result, the slave and the master have access to substantially identical layer 3 tables. In summary, during active forwarding of the master and during

standby mode of the slave, layer 3 table entries are learned by the forwarding engine of both the master and the slave from the packet header via the shared interfaces between the two routers.

As described above with reference to block 1210 of FIG. 12A, when the master fails, the forwarding engine tables are modified. More particularly, as described above with reference to FIG. 13A, the layer 2 table of the slave is modified to replace references to the master with references to the slave such that each entry is mapped to the slave rather than the master. In addition, once the slave's layer 2 table has been modified, entries associated with the master may be identified and removed from the layer 3 table so that the source of the packet is correctly identified in the packet header. In other words, the layer 3 shortcuts established by the master are purged from the layer 3 table. However, as described above, where an entry does not exist in the slave's layer 3 table for a particular path, the packet is forwarded in software. Thus, if all entries in the slave's layer 3 table that have been established by the master are removed simultaneously, a substantial amount of traffic may be forwarded in software. However, the forwarding rate in hardware is much higher than that provided in software. It is therefore desirable to delete these entries in the slave's layer 3 table gradually to reduce the traffic forwarded in software. For instance, in order to stagger the traffic handled by the CPU of the slave, the entries in the slave's layer 3 table that have been created by the master may be modified one interface/port or VLAN at a time. Subsequently, when a packet is received by the slave, a shortcut is automatically established by the forwarding engine of the slave from information provided in the packet header. The slave's forwarding engine then enters this shortcut in the slave's layer 3 table. Thus, the new entry in the slave's layer 3 table correctly identifies the slave as the source of the packet. In this manner, the shortcuts

established by the master are replaced with those established by the slave. The packet may then be switched via hardware. In this manner, the traffic handled by hardware is maximized.

When a first host wishes to communicate with a second host, it is often necessary to communicate via one or more routers. Where both hosts are directly connected to a single router, communication is accomplished through a single router or "hop." When packets must be sent via multiple routers, multiple "hops" are required. The present invention is designed to provide first hop as well as second hop routing redundancy for hosts. More particularly, when the master to slave switchover takes place, all packets from the host will be forwarded seamlessly to the destination. However, packets in the reverse direction must also be forwarded correctly even though the master has failed. This problem will be described with reference to the following figures.

As shown in FIG. 14A, a master-slave routing and switching system 1402 having a first router 1404 and a second router 1406 as described above is provided. In addition, a third router 1408 that is outside the routing and switching system 1402 is coupled to the routing and switching system 1402. As shown, the three routers 1404, 1406, and 1408 support a first VLAN 1410 and a second VLAN 1412. In the following example, a first host 1414 sends packets to a second host 1416 as well as receives packets from the second host 1416. For purposes of the following discussion, the first host 1414 has its default gateway configured with the IP address identifier and the default MAC address identifier of the default gateway. Through the default IP address identifier and the default MAC address identifier, the first router 1404 is then configured as the default gateway for the first VLAN 1410.

When the first host 1414 sends a packet to the first router 1404, the first router 1404 routes the packet to the third router 1408 to reach the final destination, the second host 1416, as shown at line 1418. Packets sent from the second host 1416 to the first host 1414 also follow the same path in the reverse direction.

When the first router 1404 fails, the second router 1406 becomes the default gateway for the first VLAN 1410 and therefore packets sent by the first host 1414 are now redirected to the second router 1406, as shown in FIG. 14B by line 1420. The second router 1406 will then route the packets to the third router 1408 which will finally forward the packets to the destination, the second host 1416. However, the reverse traffic from the second host 1416 will get forwarded by the third router 1408 to the first router 1414 since the third router 1408 has not discovered that the first router 1404 has died. Depending on the routing protocols used, the time it takes for the third router 1408 to decide that the first router 1404 has failed and to recalculate its routes varies. Moreover, this time is much greater than the time it takes for the second router 1406 (i.e., slave) to realize that the first router 1404 (i.e., master) has died according to the present invention.

To avoid "blackholing" of this reverse traffic, the traffic destined for the actual MAC address of the first router 1404 will be diverted to the second router 1406. Moreover, the second router 1406 avoids forwarding traffic back to the first router 1404. In addition, control packets destined for the first router 1404 will not be processed by the second router 1406. In this manner, reverse traffic will be forwarded by the second router 1406 (i.e., slave) and second hop redundancy is implemented.

Generally, the load sharing and redundancy technique of the present invention may be implemented on software and/or hardware. For example, it can be implemented in an

operating system kernel, in a separate user process, in a library package bound into network applications, on a specially constructed machine, or on a network interface card. In a specific embodiment of this invention, the technique of the present invention is implemented in software such as an operating system or in an application running on an operating system.

A software or software/hardware hybrid load sharing and redundancy system of this invention is preferably implemented on a general-purpose programmable machine selectively activated or reconfigured by a computer program stored in memory. Such programmable machine may be a network device designed to handle network traffic. Such network devices typically have multiple network interfaces including frame relay and ISDN interfaces, for example. Specific examples of such network devices include routers and switches. For example, the load sharing and redundancy systems of this invention may be specially configured routers such as specially configured router models 1600, 2500, 2600, 3600, 4500, 4700, 7200, 7500, and 12000 and Catalyst switches such as models 5000 and 6000 available from Cisco Systems, Inc. of San Jose, California. A general architecture for some of these machines will appear from the description given below. In an alternative embodiment, the load sharing and redundancy system may be implemented on a general-purpose network host machine such as a personal computer or workstation. Further, the invention may be at least partially implemented on a card (e.g., an interface card) for a network device or a general-purpose computing device.

Referring now to FIG. 15, a router 1440 suitable for implementing the present invention includes a master central processing unit (CPU) 1462, interfaces 1468, and a bus 1415 (e.g., a PCI bus). When acting under the control of appropriate software or firmware, the CPU 1462 is responsible for such router tasks as routing table computations and

network management. It may also be responsible for functions previously described, such as maintaining layer 2 spanning tree protocol databases, modifying forwarding engine tables of the slave router, etc. It preferably accomplishes all these functions under the control of software including an operating system (e.g., the Internetwork Operating System (IOS®) of Cisco Systems, Inc.) and any appropriate applications software. CPU 1462 may include one or more processors 1463 such as a processor from the Motorola family of microprocessors or the MIPS family of microprocessors. In an alternative embodiment, processor 1463 is specially designed hardware for controlling the operations of router 1440. In a specific embodiment, a memory 1461 (such as non-volatile RAM and/or ROM) also forms part of CPU 1462. However, there are many different ways in which memory could be coupled to the system.

The interfaces 1468 are typically provided as interface cards (sometimes referred to as "line cards"). Generally, they control the sending and receiving of data packets over the network and sometimes support other peripherals used with the router 1440. Among the interfaces that may be provided are Ethernet interfaces, frame relay interfaces, cable interfaces, DSL interfaces, token ring interfaces, and the like. In addition, various very high-speed interfaces may be provided such as fast Ethernet interfaces, Gigabit Ethernet interfaces, ATM interfaces, HSSI interfaces, POS interfaces, FDDI interfaces and the like. Generally, these interfaces may include ports appropriate for communication with the appropriate media. In some cases, they may also include an independent processor and, in some instances, volatile RAM. The independent processors may control such communications intensive tasks as packet switching, media control and management. By providing separate processors for the communications intensive tasks, these interfaces allow the master microprocessor 1462 to efficiently perform routing computations,

network diagnostics, security functions, etc.

Although the system shown in FIG. 15 is one specific router of the present invention, it is by no means the only router architecture on which the present invention can be implemented. For example, an architecture having a single processor that handles communications as well as routing computations, etc. is often used. Further, other types of interfaces and media could also be used with the router.

Regardless of network device's configuration, it may employ one or more memories or memory modules (including memory 1461) configured to store program instructions for the general-purpose network operations and other load sharing and redundancy functions described herein. The program instructions may control the operation of an operating system and/or one or more applications, for example. The memory or memories may also be configured to store routing tables, layer 2 databases, forwarding engine tables, etc.

Because such information and program instructions may be employed to implement the systems/methods described herein, the present invention relates to machine readable media that include program instructions, state information, etc. for performing various operations described herein. Examples of machine-readable media include, but are not limited to, magnetic media such as hard disks, floppy disks, and magnetic tape; optical media such as CD-ROM disks; magneto-optical media such as floptical disks; and hardware devices that are specially configured to store and perform program instructions, such as read-only memory devices (ROM) and random access memory (RAM). The invention may also be embodied in a carrier wave travelling over an appropriate medium such as airwaves, optical lines, electric lines, etc. Examples of program instructions include both machine code, such as produced by a compiler, and files containing higher

level code that may be executed by the computer using an interpreter.

Although illustrative embodiments and applications of this invention are shown and described herein, many variations and modifications are possible which remain within the concept, scope, and spirit of the invention, and these variations would become clear to those of ordinary skill in the art after perusal of this application. For instance, although the specification has described routers, other entities used to tunnel packets to mobile nodes on remote network segments can be used as well. For example, bridges or other less intelligent packet switches may also employ the standby protocol of this invention. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope and equivalents of the appended claims.